Future of Software

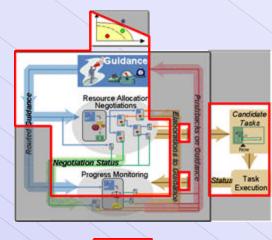
Community-Based Software Development Under Time Pressure

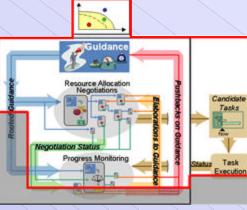
Robert Neches

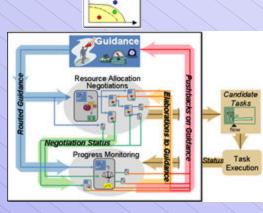
Distributed Scalable Systems
University of Southern California
Information Sciences Institute
Marina del Rey, California

Needs

- Practical applicability in days/weeks, not months/years
 - Better now beats perfect later
 - Life's a moving target
 - Gotta see it to know if we like it
 - Didn't know we wanted it before we got it
- Challenges
 - Make "build it now, fix it later" work
 - Put those pesky users in the loop





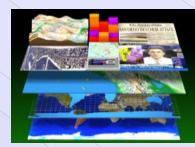


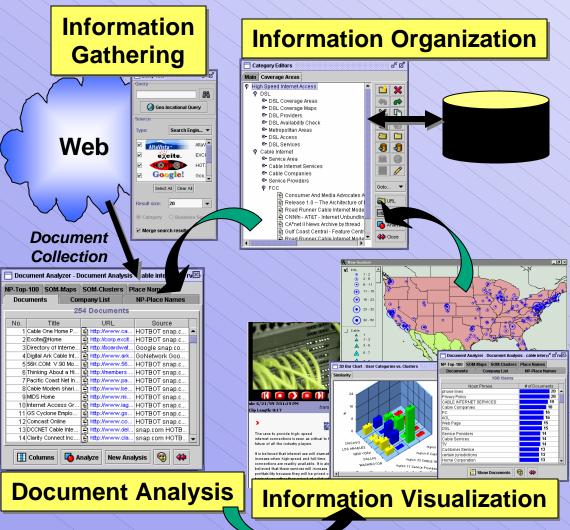
Emerging Promising Technologies

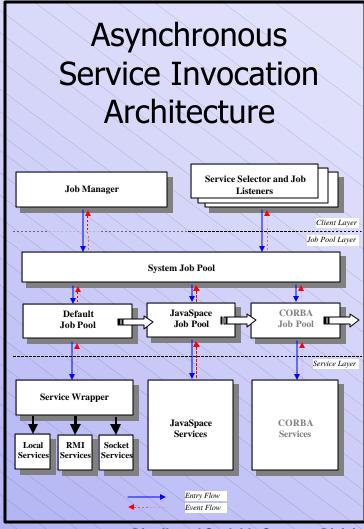
...it's not top-down structured, nor is it really bottom-up, either. It's really more like assembling a jigsaw puzzle -- you create pieces and lay them out with some blind faith that eventually you'll see how to fit them together.
-- Craig Milo Rogers

- Component-based, web-based software
- Extreme programming
- Brute force, complexity, heterogeneity
- Probes, gauges, monitors
- New insights in APIs (read the code)
- New user-centered software development tools

GeoWorlds: Component-Based Architecture

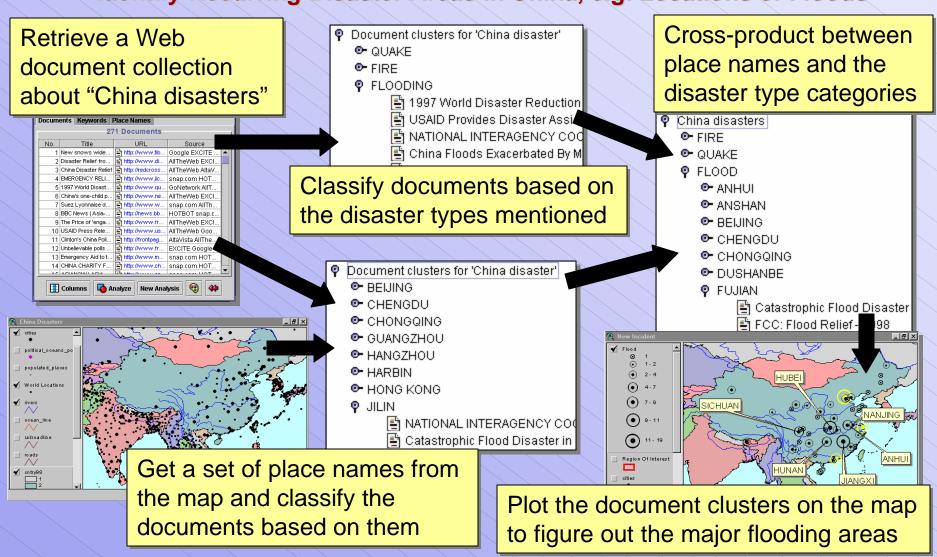




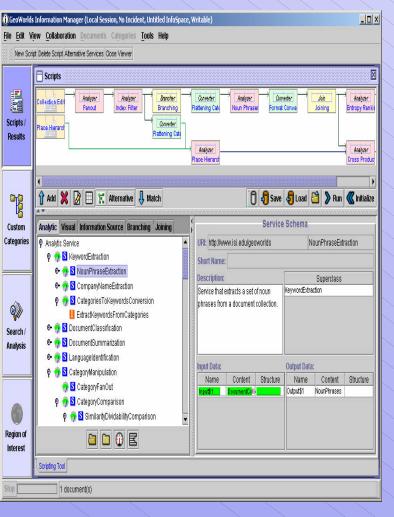


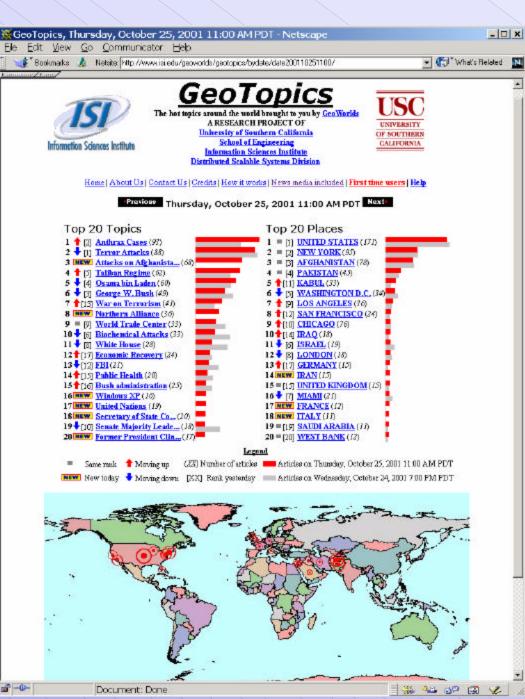
Scripting and Data Fusion (In-Young Ko)

Identify Recurring Disaster Areas in China, e.g. Locations of Floods



Large Apps in a Hurry





New Directions

...Actually, I'd rather there weren't any comments. If I want to know what it does, I'm better off reading the code.

-- Pedro Szekely

- Extraction / reverse engineering/ re-engineering tools
- Wrapper generators
- Run-time monitoring and repair
- Structure modeling and mapping
- Collaborative requirements refinement